Puzyrev V., Koric S. and Wilkin S., 2016. Evaluation of parallel direct sparse linear solvers in electromagnetic geophysical problems. Computers and Geosciences 89, 79–87. DOI:10.1016/j.cageo.2016.01.009

Evaluation of parallel direct sparse linear solvers in electromagnetic geophysical problems

Vladimir Puzyrev [a,*], Seid Koric [b], Scott Wilkin [b]

[a] *Department of Computer Applications in Science and Engineering, Barcelona Supercomputing Center, Nexus II, c/ Jordi Girona 31, Barcelona, Spain*

[b] *National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, 1205 W. Clark Av., Urbana, IL, 61801, USA*

[*] Corresponding author, vladimir.puzyrev@gmail.com

ABSTRACT

High performance computing is absolutely necessary for large-scale geophysical simulations. In order to obtain a realistic image of a geologically complex area, industrial surveys collect vast amounts of data making the computational cost extremely high for the subsequent simulations. A major computational bottleneck of modeling and inversion algorithms is solving the large sparse systems of linear ill-conditioned equations in complex domains with multiple right hand sides. Recently, parallel direct solvers have been successfully applied to multi-source seismic and electromagnetic problems. These methods are robust and exhibit good performance, but often require large amounts of memory and have limited scalability. In this paper, we evaluate modern direct solvers on large-scale modeling examples that previously were considered unachievable with these methods. Performance and scalability tests utilizing up to 65 536 cores on the Blue Waters supercomputer clearly illustrate the robustness, efficiency and competitiveness of direct solvers compared to iterative techniques. Wide use of direct methods utilizing modern parallel architectures will allow modeling tools to accurately support multi-source surveys and 3D data acquisition geometries, thus promoting a more efficient use of the electromagnetic methods in geophysics.

Keywords: Numerical modeling; linear systems; direct solvers; parallel computing; controlled-source electromagnetics; geophysical exploration.

1. INTRODUCTION

Geophysical exploration techniques aim to determine various physical properties both in the shallow and deep interior of the Earth. Electromagnetic (EM) tools are applied in different areas of geophysics such as hydrocarbon and mineral exploration, geothermal reservoir monitoring, CO2 storage imaging to name a few. These techniques utilize low-frequency EM energy to map variations in the subsurface electrical conductivity and characterize the geological structure at depths ranging from a few tens of meters to several kilometers. Recently emerged controlled-source electromagnetic (CSEM) imaging technologies in marine and land environments are receiving special attention today. Marine CSEM method has become a common tool in industry for reducing ambiguities in data interpretation, reducing exploratory risk and its small environmental footprint because of its unique ability to reliably extract information on fluid properties of the geological structures. Land-based EM techniques are often applied for reservoir monitoring, while airborne methods have found wide use in hydrogeological investigations. The academic and industrial developments of various EM technologies are reviewed in Chave and Jones (2012), Constable (2010), Key (2012), Pellerin (2002), Siemon et al. (2009) and Streich (2015).

The modeling and inversion of the three-dimensional EM data in the frequency domain requires considering many sources and frequencies. The computational cost is enormous and the use of high performance computing (HPC) is mandatory. Deriving a solution requires 80–90% of the total computational time be spent calculating thousands of large sparse systems of linear equations. The progress in numerical mathematics and parallel computing during the last decades has permitted using direct methods for the large linear systems with millions of unknowns. Direct solvers are used in many academic and industrial fields, such as condensed matter physics, fluid dynamics, biomechanics, structural analysis, aerodynamics as well as many others. These methods are especially efficient for multisource problems since the expensive matrix factorization part of the solution needs to be performed only once, and then multiple solutions are rapidly obtained by relatively low cost back solve operations. In this context, parallel direct solvers have been successfully applied in seismic (Operto et al., 2007; Brossier, 2011) and EM (Blome et al., 2009; Streich, 2009) modeling and inversion.

Several evaluation studies have been published on direct solver performance including sequential tests (Gould et al., 2007), early evaluation of two distributed-memory solvers (Amestoy et al., 2001) and recent massively parallel tests for real-valued systems from solid mechanics (Koric el al., 2014). In geophysical modeling publications, usually one type of solver is used and no comparative analysis is given (with a few exceptions, such as Pardo et al., 2012 and Kordy et al., 2016). There is not a comparison in the existing literature of direct solvers on modern multi-

core architectures applied to complex linear systems with many millions of unknowns arising from geophysical applications.

In this paper, we present evaluation and comparison of several state-of-the-art parallel direct solvers on one of the world's most powerful supercomputers. The complex linear systems arise from different electromagnetic modeling problems in geophysical exploration. The remainder of the paper is organized as follows. In Section 2 we give the necessary background on the physical problem and numerical approaches and iterative and direct linear solvers are described in Section 3. In Section 4 we analyze the dependence of computational and memory demands on the number of unknowns in the linear system and determine the largest problems that can be solved on modern shared-memory and distributed-memory platforms. We show large-scale multisource modeling examples that previously were considered unfeasible with direct solvers and report scalability in Section 5. The maximum performance achieved was 97 TFLOPS on 65 536 cores resulting in, to our knowledge, a new record in sparse matrix factorization. Section 6 contains concluding remarks and a brief description of our plans to apply the outcomes of this work in the near future.

2. MODELING APPROACHES

Electromagnetic field behavior at low frequencies used in geophysics is governed by the diffusive Maxwell's equations that in the frequency domain are (Jackson, 1999)

$$\nabla \times \mathbf{E} = i\omega\mu\mathbf{H}, \quad \nabla \times \mathbf{H} = \mathbf{J} + \sigma\mathbf{E}. \tag{1}$$

Here $\mathbf{E}$ is the electric field, $\mathbf{H}$ is the magnetic field, ω is the angular frequency and $\mathbf{J}$ is the electric current source. It is common in geophysics to eliminate $\mathbf{H}$ by substituting it from the first equation of (1) into the second, obtaining the second-order partial differential equation for the electric field

$$\nabla \times \nabla \times \mathbf{E} - i\omega\mu\sigma\mathbf{E} = i\omega\mu\mathbf{J}. \tag{2}$$

For Earth materials, the magnetic permeability μ is usually assumed to be equal to the constant free space value μ_0. Dielectric permittivity can be safely neglected at typical frequencies used for geophysical exploration making the electric conductivity σ the main characteristic of the medium. Electrical anisotropy is taken into account by treating it as a transversely-isotropic tensor $\bar{\sigma}$. Generally speaking, air is insulating ($\sigma = 0$), but in numerical modeling its conductivity is usually set to some small value without significant loss of accuracy (see Section 3). In order to avoid the source singularities and reduce the air layer effect, the electric field is often split into the

sum of the primary and secondary field contributions $\mathbf{E} = \mathbf{E}_p + \mathbf{E}_s$. In this case, the governing equation becomes

$$\nabla \times \nabla \times \mathbf{E}_s - i\omega \mu_0 \bar{\sigma} \mathbf{E}_s = i\omega \mu_0 (\bar{\sigma} - \bar{\sigma}_p) \mathbf{E}_p . \tag{3}$$

After this equation is solved, the magnetic field can be interpolated from the electric field values using Faraday's law. This formulation is convenient and simple, and is used in the majority of industrial modeling codes. The main disadvantages of this approach are the numerical problems caused by the nullspace of the curl operator (Hiptmair, 1998) and lower accuracy of the magnetic field components that are interpolated from the electric ones. Instead of solving the electric field equation we can use the common practice of reformulating the problem in terms of vector and scalar potentials $\mathbf{V}, \Phi$ (Biro and Preis, 1989; Haber et al., 2000; Jahandari and Farquharson, 2015). In this case, the following system consisting of a vector and a scalar equation is to be solved

$$\nabla^2 \mathbf{V}_s + i\omega\mu_0\bar{\sigma}(\mathbf{V}_s + \nabla\Phi_s) = -i\omega\mu_0\bar{\sigma}_\Delta(\mathbf{V}_p + \nabla\Phi_p), \tag{4.1}$$

$$\nabla \cdot [i\omega\mu_0\bar{\sigma}(\mathbf{V}_s + \nabla\Phi_s)] = -\nabla \cdot [i\omega\mu_0\bar{\sigma}_\Delta(\mathbf{V}_p + \nabla\Phi_p)] . \tag{4.2}$$

Both formulations (3) and (4) are considered in this evaluation study. The current numerical modeling techniques applied to electromagnetic problems in geophysics are the finite-difference (FD) and the finite-element (FE) methods, although other methods such as integral equations have also received widespread use. The FD method was the most popular numerical technique in EM modeling during the last two decades (Rodi and Mackie, 2001; Siripunvaraporn et al., 2005; Commer and Newman, 2008; Kelbert et al., 2014) because of its relatively simple formulation, convenient grid building and ease of adaptation to new computer architectures. FE schemes support completely unstructured meshes that allow mesh refinements (Figure 1) and make irrelevant the geometrical complexity of topography or bathymetry. They are particularly flexible in adapting to the shape of geological bodies and thus allow for highly accurate numerical simulations (Zehner et al., 2015). The discretization with either of the above-mentioned methods results in a large sparse system of linear equations with complex numbers for each frequency and source position

$$\mathbf{A}\,\mathbf{x} = \mathbf{b} . \tag{5}$$

This paper assumes that the complex matrix $\mathbf{A}$ is symmetric, non-Hermitian and indefinite. Its sparsity structure is determined by the underlying discretization method and grid type. The matrix obtained from Eq. (3) using the FD method and structured grids is 13-diagonal banded, while discretization of Eq. (4) for the unstructured FE meshes leads to a general sparsity pattern. The matrix $\mathbf{A}$ is highly ill-conditioned for most realistic cases.

A geophysical forward modeling code can be used as a standalone program for survey planning, feasibility study and academic purposes, though it often serves as a part of an inversion framework. In this case, every iteration of the inversion algorithm involves the solution of multiple systems (5). The computation of the data misfit gradient (Tarantola, 2005), in turn, leads to the adjoint problem that requires the solution of the same number of systems with matrix **A** transposed and different right-hand-sides (RHS). If a second-order minimization scheme such as Gauss–Newton is employed, additional solves would be required for each Hessian matrix–vector product. A total number of linear systems to be solved can be estimated as

$$N_{iter} \times C \times N_f \times N_s , \tag{6}$$

where N_{iter} is the number of iterations required to converge (a few tens to hundreds), C is the constant depending on the minimization algorithm (usually 3–5), N_f and N_s are respectively, the number of frequencies and sources (or receivers considering reciprocity) in the survey. N_s typically, is in the order of tens to hundreds for land CSEM studies, while it reaches thousands in modern large-scale marine CSEM surveys and airborne electromagnetics (Siemon et al., 2009; Newman, 2014). Consequently, the whole inversion of a large dataset using the traditional approach may require a solution of up to several millions of complex linear systems each having millions of unknowns. There is a current trend to combine various EM techniques or use them together with the seismic methods that resolve geological structure with higher resolution and can constrain the interpretation of the EM measurement (Hu et al., 2009). The differences between the constrained and unconstrained inversions are shown in Figure 2. In this case, we aim to locate as accurate as possible the thin resistive body that may be a hydrocarbon reservoir. Due to the diffusive nature of the electromagnetic field, standalone EM inversion produces a very blurred picture. The anomaly is clearly visible, but its position and dimensions are not precise. When the target boundaries are known from the seismic data, EM inversion can be constrained by these results leading to a much better quality of the restored model. A focused image of a target with sharp boundaries can be achieved, for example, by the incorporation of prior information about known geological structures within the regularization functional (Newman and Hoversten, 2000; Zhdanov, 2009). In this way, one can preserve known structural boundaries within the inversion domain where rapid changes in the conductivity are allowed, and impose smoothness constraints elsewhere. On the contrary, this may introduce an apparent resolution into resistivity images which cannot be justified by the inherently diffusive nature of low-frequency EM fields. Either this scheme or the so-called joint inversion process (when seismic and electromagnetic parameters are inverted simultaneously) brings an accelerated computational challenge.

Modeling different source positions in one survey can be performed in several ways. A local grid for each source greatly minimizes the dimensions of the problem, however, at the cost of interpolation of all data between different grids in the inversion process (Commer and Newman, 2008). Since the FE method supports unstructured tetrahedral meshes that allow for complicated geometries and arbitrary mesh refinements, one mesh can be created for all sources. Our experiments show that adding several tens or hundreds of source/receiver positions to the original tetrahedral mesh does not lead to a large increase in the number of elements. This approach has been applied routinely to the conventional FD codes that use structured orthogonal grids and hence are less flexible in local mesh refinements. The main reason for creating a global mesh is its effectiveness with a direct solver used for the solution of (5), i.e. performing one matrix factorization and re-using it for forward and adjoint modeling with all sources. This allows to reduce the cost of forward modeling within an inversion from (6) to approximately $N_{iter} \times N_f$ tasks (each task now includes a matrix factorization and multiple solves). Also, from the computational point of view, direct solvers are advantageous with the second-order inversion methods, such as Newton or Gauss–Newton, which converge in much less iterations than the first-order methods (Pratt et al., 1998).

3. LINEAR SOLVERS

3.1. Direct and Iterative Methods

The choice of a robust and efficient solver is critical for the performance of a frequency-domain modeling code. Linear solvers can be divided into two broad groups: iterative and direct methods. In 3D problems, the scale of the geometry required to give meaningful results has, until recently, been accessible only to iterative solvers such as Krylov subspace methods because of their low memory and computational requirements. Robustness of iterative methods is dependent on the condition number of the system matrix and hence they require the utilization of robust preconditioners in real-size applications. At the end of the 20th century many iterative solvers and preconditioning techniques were proposed; see Barrett et al. (1994), Saad (2003) and references therein. In some cases, a problem specific preconditioner will be highly effective, yet it is often difficult to parallelize the code without losing either its efficiency or scalability on modern HPC platforms. Koric et al. (2014) has recently explored several configurations of solvers, preconditioners, and their parameters in the libraries PETSc and hypre and found that no iterative preconditioned solver combination could correctly solve the highly ill-conditioned system of equations from the structural finite element simulations.

Nearly a decade ago, two things changed in ways that transformed the landscape for sparse linear solvers making room for other methods in geophysical modeling codes. As explained in the previous section, the increasing number of source and receiver positions in CSEM methods became a serious computational challenge. Conversely direct solvers that had been considered impractical before in 3D cases, became feasible in realistic problems. These realistic problems show the main benefit from the computational point of view in the run-time efficiency for multisource problems. If a single mesh can include all source positions of the survey, (5) becomes a system with multiple RHS. In this case, the number of sources has a smaller impact on the computation time of a direct solver compared to the cost of the matrix factorization. Direct methods while avoiding convergence problems of iterative solvers, they are much more expensive in terms of memory due to fill-in in triangular factors. A wider use of direct solvers on parallel systems was also restricted by the fact that some packages suffered from a poor computation-to-communication ratio.

The most widely used direct methods are variants of Gaussian elimination and perform the solution of sparse systems of linear equations by factorizing the coefficient matrix. Several types of factorization can be computed depending on the matrix properties (Golub and Van Loan, 1996)

$$\mathbf{A} = \mathbf{L}\,\mathbf{L}^T \tag{7.1}$$

$$\mathbf{A} = \mathbf{L}\,\mathbf{D}\,\mathbf{L}^T \tag{7.2}$$

$$\mathbf{A} = \mathbf{L}\,\mathbf{U} \tag{7.3}$$

The factorization (7.1) is known as the Cholesky factorization and is available only for symmetric positive definite matrices. Our case of symmetric indefinite matrices allows for a $\mathbf{LDL}^T$ factorization (7.2) that's speed performance outpaces the general $\mathbf{LU}$ decomposition (7.3). Once the factorization is computed, two additional triangular solves are needed to obtain the solution, for example, a forward solve $\mathbf{LDy} = \mathbf{b}$ and a backward solve $\mathbf{L}^T\mathbf{x} = \mathbf{y}$. This phase requires considerably less computational time than previous factorization.

The condition number of the system matrix is greatly affected by the chosen conductivity of the air layer σ_{air}. Since the convergence of iterative methods may stagnate for very small (more realistic) values of σ_{air}, in first 3D applications it was chosen as 10^{-4} or 10^{-5} S/m. These values may lead to inaccurate modeling results, especially for high frequencies. For low frequencies the first term in Eq. (2) completely dominates the second term in the regions with very low conductivity (i.e. air), making the iterative solution inaccurate unless a special technique is used for numerical stabilization (Demkowicz and Vardapetyan 1998). When using an iterative method, the researcher often has to compromise accuracy of the solution with the computational effort. Direct

solvers typically do not have these limitations and we can choose σ_{air} as 10^{-8} or 10^{-10} S/m ensuring the absence of any numerical errors caused by the unrealistic representation of the air.

3.2. Solvers used in this study

For this evaluation we have selected three direct solver packages listed in Table 1. Our selection criteria included a parallel implementation for modern architectures, continuous development, support of the code, and a wide range of sparse matrix types and factorization/ordering algorithms supported. The last point is important from a computational point of view, since even though a general factorization algorithm can be applied to a symmetric matrix, a specific symmetric factorization will be performed much faster. For example, a popular solver SuperLU_DIST (Li and Demmel, 2003) with recently added OpenMP and CUDA support does not take advantage of matrix symmetry and consequently was not included in this study. We have also excluded solvers that can be used only as parts of more general application software. Some of the packages in Table 1 are freely available to academic users in the full version, while it is necessary to purchase a license for the others.

As it can be deduced from its name, MUMPS ("MUltifrontal Massively Parallel Solver") by Amestoy et al. (2001, 2006) is a distributed-memory multifrontal package for solving sparse systems of linear equations with the matrix that can be either unsymmetric, symmetric positive definite, or general symmetric. MUMPS is implemented using a fully asynchronous approach with dynamic scheduling of the computational tasks. It supports real and complex systems in single or double precision, and is able to compute a Schur complement matrix, determinant and condition number estimates. The parallel version of MUMPS requires MPI, BLAS, BLACS, and ScaLAPACK libraries to be installed on the system.

The shared-memory solver PARDISO (Schenk and Gärtner, 2004; Kuzmin et al., 2013) uses a combination of left- and right-looking supernode techniques for factorization. It supports a wide range of sparse matrix types and computes the solution of real or complex, symmetric, structurally symmetric or nonsymmetric, positive definite, indefinite or Hermitian sparse linear systems on shared-memory multiprocessing architectures. A new MPI-based version for distributed-memory architectures has been implemented in PARDISO 5.0, however it is only available for real symmetric indefinite matrices. In this study we were only able to evaluate in a shared-memory environment.

The Watson Sparse Matrix Package, WSMP (Gupta et al., 1998; Gupta, 2002), is a distributed-memory multifrontal solver with hybrid MPI/Pthreads parallelization. It comprises two parts, one for solving symmetric systems (with positive-definite, quasi-definite, and indefinite

matrices, with or without diagonal pivoting) and one for general systems (Gupta, 2007). WSMP was primarily developed as a highly scalable parallel code and allows using multiple MPI processes on one shared-memory node; the best performance is observed with 2–8 threads per MPI processes. The parallel symmetric factorization is based on the multifrontal algorithm described in Liu (1992).

Solving a sparse linear system includes three main tasks: ordering and symbolic factorization, numerical factorization and the solution (forward/backward substitutions and optionally iterative refinement). When **A** is sparse, the triangular factors **L** and **U** typically have nonzero entries in many more locations than **A** does. This phenomenon is known as fill-in, and results in increased memory and time requirements of a direct method to solve a sparse system with respect to the size of the system. During the first phase, an ordering algorithm is typically applied to minimize fill-in during factorization and to provide more parallelism. PARDISO can use either the nested dissection algorithm from the METIS package (Karypis and Kumar, 1998) or the minimum degree algorithm. WSMP by default uses graph-partitioning based ordering techniques and has an option to use a minimum local fill ordering. MUMPS has, perhaps, the widest choice of ordering options, including several variants of minimum degree algorithms, minimum fill, PORD, METIS and SCOTCH. The proper choice of ordering algorithm is crucial due to the large impact on the performance (Pardo et al., 2012). For example, METIS-based ordering, which is perhaps the most widely used with MUMPS, was confirmed as superior to the default option. In order to make this evaluation objective, in the tests below, we have used the METIS-based ordering for MUMPS and the default ordering algorithms for PARDISO and WSMP. In symbolic factorization, the precomputation of the nonzero pattern of the factors helps to predict the memory requirements of numerical factorization as the distributing of data and computation among processing units in a parallel implementation.

In utilizing a direct method solver, the largest cost is matrix factorization. The factorization can be performed in several different ways, offering a trade-off between efficiency, parallelism and memory usage. The established techniques are left-looking, right-looking, and multifrontal algorithms (Duff and Reid, 1983; Liu, 1992; Gould et al., 2007). For very large matrices, special versions of the solvers are available that use 8-byte integers. When the system is so large that the factors cannot fit into main memory, some direct solvers provide an out-of-core (OOC) mode. OOC mode writes and reads from secondary storage are typically much slower than working with the main memory (Avron and Gupta, 2012), so we did not consider OOC mode in this paper. The main objective of this study is to investigate how different parallel implementations can take advantage of large amounts of distributed memory, multicore processors and low latencies and

increased bandwidth of modern interconnect network technologies.

4. NUMERICAL EXAMPLES

Full-scale performance and scalability tests of shared-memory direct solvers have been performed up to 65 536 cores of the sustained petascale system of Blue Waters. Capable of a peak performance of 13.3 PFlops, Blue Waters at University of Illinois' National Center for Supercomputing Applications, USA is one of the most powerful supercomputers in the world. It combines 22 640 XE compute nodes (two 8-core AMD 6276 processors and 64 GB of memory each) with 4 224 XK nodes (one 6276 processor and one NVIDIA Kepler K20X GPU accelerator with 2 688 cores; 32/6 GB of memory), all connected by the Cray Gemini torus interconnect. The solvers were linked with AMD ACML math library for Basic Linear Algebra Operation (BLAS) operations and take full advantage of increased memory bandwidth and SSE instructions of XE6 architecture.

4.1. Test matrices

For this study, we have chosen four test systems resulting from large-scale EM problems. Two independent FE and FD modeling codes (Puzyrev et al., 2013, 2015) used for the simulations were developed in Barcelona Supercomputing Center. Table 2 summarizes the test matrices varying from 1 to 8 million degrees of freedom (DOFs). The condition number, as a direct indicator of ill-conditioning, is driven by several factors, including irregular element shapes, conductivity contrasts between the materials, high anisotropy and realistic air conductivity. M5 represents a complex FE case with many refinement positions in the mesh (size of elements ranging from 3 m to 3000 m at the boundaries) and hence a very high condition number. M8 is also a challenging example for direct solvers whose condition number cannot be even estimated with high degree of confidence, so we provided an approximate value. Both these matrices are built from the complex model shown in Figure 6 of Puzyrev and Cela (2015). The original model was modified to accommodate up to 1000 receiver positions and to have higher conductivity contrasts, anisotropy factor of up to 10 and air conductivity of 10^{-10} S/m.

Given the computational complexity requirements for time and memory of a sparse numerical factorization are superlinear, and since the computation of the solution of a sparse linear system of n equations with complex coefficients is equivalent to the computation of the solution of a sparse linear system of $2n$ equations with real coefficients, we typically find direct solvers are applied to complex systems constrained to less than several millions of unknowns. This is

especially true for matrices resulting from the FE method that are much denser than FD ones. Recently, Operto et al. (2014) reported the case of 6.7 millions of unknowns in 3D acoustic FD seismic modeling. After analyzing the existing literature, we believe the size of the two largest complex systems in this study stands as the largest ever in geophysics to be benchmarked with direct solvers, for both FE and FD methods.

4.2. Performance in a shared-memory environment

First, we evaluate the performance of direct solvers on medium-scale problems. Tables 3 and 4 compare the wall clock times and memory demands for solving the M1 and M2 systems, respectively. For comparison, the symmetric iterative solver QMR (Freund, 1992) with the preconditioner described in Puzyrev et al. (2013) was used in these tests. The computations were made on a single Xeon E5-2670 processor with 8 cores and equipped with 128 GB of memory.

As we can see, matrix factorization is the most costly part of the solution for all solvers. Iterative method, expectedly, has very small memory requirements and a low cost preprocessing phase, but a large time-per-RHS ratio. Hence, it can be competitive in terms of computational time only when the number of sources does not exceed a few tens. MUMPS and WSMP required more memory than the other codes which is typical for multifrontal methods. Differences in the computational time and memory demand can be explained by the implementation details or ordering of the unknowns. PARDISO is based on memory-efficient supernodal techniques, requiring lower amounts of memory. As a result, for some test matrices, PARDISO was able to perform the factorization using one computational node, while WSMP and MUMPS required distributing the task among several nodes. However, as we see from Tables 3 and 4, PARDISO is a bit slower than MUMPS on the FE matrix and much slower than WSMP on the FD case.

Of our test matrices listed in Table 2, the shared-memory solver PARDISO was only capable in solving the M1 and M2 due to insufficient memory in other cases. Due to this constraint, we tested PARDISO on a shared-memory machine with an additional set of linear systems, which allowed us to gradually increase the problem size. We use a sequence of matrices derived from three-dimensional FD grids for the same model but different frequencies, which is typical for frequency domain EM problems. The number of cells in each dimension is usually based on the skin depth criteria (Constable, 2010) and is proportional to $\sqrt{f}$, while the total number of unknowns in three dimensional problems grows as n^3. In this example, each subsequent frequency is 25% larger than the previous one, leading to a ~40% increase in the number of system unknowns. Figure 3 shows the results of a PARDISO performance study on a shared-memory machine with 16 cores and 128 GB of memory. This solver was able to solve linear systems with

up to 3.9 million unknowns. The memory demand in the largest case was 116 GB. Only 10-15% of the machine peak performance was achieved. The computational time has a nonlinear growth proportional to $O(n^{2.1})$ of the number of system unknowns, which is close to the theoretical complexity of a sparse numerical factorization in 3D of $O(n^2)$. The growth of the memory demand is close to $O(n^{1.4})$ versus the theoretical of $O(n^{4/3})$.

In some applications, such as magnetotelluric modeling, the same grid is often used for a set of frequencies. This leads to a sequence of matrices of the same size and structure, but with different entries. The matrix condition number is usually increasing with decrease in frequency, leading to very ill-conditioned systems for low frequencies. Another example of the robustness of direct solvers in shown in Figure 4 where we measure factorization time for a system of fixed size (1 million of unknowns) for a range of decreasing frequencies (i.e. increasing condition numbers). As it can be seen, factorization time is independent from the system matrix condition number in this case.

Finally, we study the many-RHS performance of the solvers under consideration that is of a great interest for multisource CSEM problems. As shown in Figure 5, actual runtimes deviate a bit from the times extrapolated from a single RHS. This is caused by the blocking strategy of the solvers when the matrix factors are accessed only once for each block of RHS. In the case of the M1 system, the computational effort for 1000 forward and backward substitutions is roughly comparable to the cost of the matrix factorization, however, the exact relation between these values depends on many parameters. Default values for the block size parameters were used in the solvers' setup.

5. SCALABILITY

In this section, we present a strong scalability study of the MUMPS and WSMP solvers. Since the factorization algorithms are bounded by relatively low computation-to-communication ratio, it is generally believed in the modeling community that direct solvers are best suited for a relatively small numbers of distributed nodes, while iterative methods are preferred for highly scalable simulations (Sourbier et al. 2011). Conversely, this study has shown very promising results on direct solvers' scalability. Even though distributed-memory solvers encounter communication and load-imbalance overheads, they use the aggregate memory bandwidth of all computational nodes. For solvers like PARDISO that can run only on several shared-memory cores, the scalability is limited by the total memory bandwidth of a single node.

For each test case, we start the benchmarking with the minimum number of nodes that can

fit the matrix factorization. Parallel speedup is defined as the ratio of wall clock time on the minimum number of cores over wall clock time on p cores. To utilize the floating point units on XE6 nodes, we spawn two MPI ranks per node each with 8 threads. Figure 6 compares the scalability of WSMP and MUMPS solvers for the solution of the M2, M5 and M8 test systems. $\mathbf{LDL}^T$ factorization wall clock time on the logarithmic scale is plotted on the vertical axis. With the exception of a single case (matrix M5 on 64 cores), we see that WSMP is much faster than MUMPS, especially for the FD system. For a relatively small 3D problem such as M2, MUMPS reached its scalability limit at 128 cores, WSMP scaled well until 1024 cores. In this case, EM inversion on a large supercomputer can benefit substantially from an additional parallelization over frequencies. The performance of MUMPS for the FE system M5 on 64 cores is similar to WSMPs' performance. However, MUMPS parallel performance quickly deteriorates as the number of cores increases and consequently more data is exchanged between the MPI processes. WSMP shows excellent parallel scalable behavior even when the number of cores reaches into the thousands. It took approximately 1 minute to factor M5 matrix on 16 384 cores. This opens the possibility for solving large-scale FE problems very efficiently.

Both M5 and M8 systems are highly ill-conditioned and practically solvable only with direct methods. An iterative solver with a specially-tailored powerful preconditioner can also be used, however all the combinations of solvers, preconditioners, and their parameters that we considered failed to converge to the desired tolerance for these systems after many thousands of iterations. MUMPS also failed to perform the factorization of M8 running to the wall clock limit of 6 hours without making any progress in factorization, while WSMP successfully finished all tasks. On the largest number of cores, the factorization of this huge complex M8 matrix was performed in 30 seconds! This confirms the speed and the robustness of direct methods and their small sensitivity to the modeling frequency and conductivity distribution in the model. The scalability gradient of WSMP remains high even on 65 536 cores indicating that its scalability limit has not been reached. We might also note that the factorization of the M8 system with WSMP was performed even faster than of the M5 case, which can be explained by sparsity structure and larger number of nonzeros in the FE matrix.

While the memory usage per node decreases, the total memory usage grows when more than one computational node is used. For example, when two supernodes are factored concurrently, both frontal matrices should be kept in main memory simultaneously (Avron and Gupta, 2012). Thus, by exploiting more parallelism, we also increase the total memory requirements. On some systems one needs to find a good balance between the benefit of using more parallelism and the cost of additional data movements in the memory. Figure 7 shows the memory usage for the

factorization of the largest systems that can be solved with the distributed-memory solvers (M8 for WSMP and M5 for MUMPS). When the number of nodes is small, factorization of these huge complex matrices barely fit into the system memory. On 4 096 nodes (65 536 cores) the memory usage significantly drops to 3.8 GB/MPI rank (15.2 GB/node) showing that even systems larger than M8 should be solvable with WSMP on Blue Waters or other similar modern large HPC systems.

Parallel speedup of the factorization of the three largest test matrices is shown in Figure 8. We see that on low number of cores almost all cases experience an enhanced speedup that is even super-linear for the M5 and M8 systems with WSMP. On these scales, computation is more important than communication and since more cache is available with more nodes, more data is stored to cache rather than to memory. Thus, the memory access time is dramatically reduced, which causes the extra speedup in addition to that from the actual computations. Scaling wider results in more communications and synchronization overheads, especially for the small M2 problem. On the largest model M8, WSMP is able to keep the speedup close to ideal until it reaches 8 192 cores. From this number, the speedup growth rate slowly begins to saturate. These results show that a hybrid parallel implementation in WSMP can well-utilize a petascale HPC system and provide good factorization time and parallel scalability performance. Last but not least, this is achieved due to the Gemini interconnect between XE6 nodes of Blue Waters having lower latency and better bandwidth.

Regarding the differences in the performance of WSMP and MUMPS, we suspect that it is due to WSMP's capability of utilizing an optimally scalable parallel factorization algorithm (Gupta et al., 1997). In this multifrontal algorithm all frontal and update matrices are distributed among groups of MPI processing using a block-cyclic two-dimensional partitioning that seems to be a key to a highly scalable formulation. MUMPS, on the other hand, uses a one dimensional partitioning of all frontal matrices with the exception of the root of the elimination tree (i.e. only this root supernode is partitioned in two dimensions). Numerical examples in this paper also illustrate scalability advantages of the two-dimensional partitioning.

The number of floating point operations (FLOPs) to perform matrix factorization is constant for a particular solver and particular test system. Figure 9 summarizes the observed performance for all scaling runs showing the hardware FLOPs for MUMPS and WSMP. The performance FLOPs/sec is greatly increasing as we scale wider, since the factorization wall clock time is decreasing. For the M8 matrix, we achieved the highest performance on Blue Waters of 97.05 TFLOPS using 65 536 cores. To our knowledge, this is the record performance in a sparse matrix factorization using direct solvers.

6. CONCLUSIONS AND FUTURE WORK

The recent rise of terascale and petascale computing, as well as the developments in multifrontal and supernodal algorithms for matrix factorization, has greatly increased the efficiency and practicality of direct solvers. We have evaluated the performance of modern parallel shared- and distributed-memory direct solvers for solving large sparse complex linear systems arising from electromagnetic problems in geophysics on the Blue Waters sustained petascale system. The test systems were extracted via two practical finite-element and finite-difference codes that are state-of-the-art solutions for the most challenging problems in exploration geophysics today. Computational demands, parallel scalability and robustness of three direct solvers were evaluated on large complex-valued systems of linear equations. The growth rate of the computational time and memory depending on the number of system unknowns was found to be close to the theoretical values. Experimental results show that WSMP is significantly faster on large systems, more scalable and thus can take full advantage of a modern high performance computing system.

In our future work, we plan to exceed the limit of 10 millions of complex unknowns with a direct solver. Since substantial efforts have been undertaken to adapt linear sparse solvers for evolving GPU accelerated heterogeneous systems, we plan to test the newly accelerated WSMP port on the Cray XK7 nodes of Blue Waters. Direct methods, while being extremely efficient for multisource problems, have a small sensitivity to various model parameters such as geometry of the domain, modeling frequency or air conductivity. By using high-quality structured or unstructured grids, the electromagnetic field behavior can be modeled with extreme fidelity helping to improve the understanding of the subsurface. Having multiple transmitter positions is crucial for proper illumination of reservoirs and for the quality of inversion. In these experiments, WSMP reached a performance of 97 TFLOPS on 65 536 cores of Blue Waters in the solution of the linear system with 8 million unknowns. Our scalability study credibly shows that this direct solver is prepared for even larger HPC simulations, thus enabling extremely realistic and robust simulations of the 3D electromagnetic exploration problem. Future development lines in geophysical exploration will tackle the joint inversion, and in particular when electromagnetic soundings are carried out together with seismic exploration. Joint inversion of both types of data will lead to a massive leap in terms of understanding the properties of subsurface and hence having crystal-clear pictures of the Earth's interior, which would be a major breakthrough in geophysics for the foreseeable future.

ACKNOWLEDGMENTS

The authors would like to thank the MUMPS and PARDISO developers for providing free

academic licenses and Dr. Anshul Gupta for access to his solver library which is not publicly available. We also would like to thank the Private Sector Program and the Blue Waters sustained-petascale computing project at the National Center for Supercomputing Applications (NCSA), which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois. The shared-memory tests were performed on the MareNostrum supercomputer of the Barcelona Supercomputer Center. The first author acknowledges funding from the Repsol-BSC Research Center through the AURORA project and support from the RISE Horizon 2020 European Project GEAGAM (644602). The authors wish to thank Jef Caers and two anonymous reviewers for their valuable comments that significantly helped to improve this paper.

REFERENCES

Amestoy, P. R., Duff, I. S., L'Excellent, J.-Y., Li, X. S., 2001. Analysis and comparison of two general sparse solvers for distributed memory computers. ACM Trans. Math. Software 27(4), 388–421.

Amestoy, P. R., Guermouche, A., L'Excellent, J.-Y., Pralet, S., 2006. Hybrid scheduling for the parallel solution of linear systems. Parallel Comput. 32(2), 136–156.

Avron, H., Gupta, A., 2012. Managing data-movement for effective shared-memory parallelization of out-of-core sparse solvers. In High Performance Computing, Networking, Storage and Analysis (SC), 2012 International Conference Proceedings, 1–11.

Barrett, R., Berry, M. W., Chan, T. F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., Van der Vorst, H., 1994. Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods. SIAM, 142 pp.

Biro, O., Preis, K., 1989. On the use of the magnetic vector potential in the finite element analysis of three-dimensional eddy currents. IEEE Trans. Magn. 25, 3145–3159.

Blome, M., Maurer, H. R., Schmidt, K., 2009. Advances in three-dimensional geoelectric forward solver techniques. Geophys. J. Int. 176, 740–752.

Brossier, R., 2011. Two-dimensional frequency-domain visco-elastic full waveform inversion: Parallel algorithms, optimization and performance. Comput. Geosci. 37(4), 444–455.

Chave A. D., Jones, A. G., 2012. The Magnetotelluric Method: Theory and Practice. Cambridge University Press, 570 pp.

Commer, M., Newman, G. A., 2008. New advances in three-dimensional controlled-source electromagnetic inversion. Geophys. J. Int. 172, 513–535.

Constable, S., 2010. Ten years of marine CSEM for hydrocarbon exploration. Geophysics 75(5), A67–A81.

Demkowicz, L., Vardapetyan, L., 1998. Modeling of electromagnetic absorption/scattering problems using hp-adaptive finite elements. Comp. Meth. Appl. Mech. Eng. 152, 103–124.

Duff, I. S., Reid, J. K., 1983. The multifrontal solution of indefinite sparse symmetric linear equations. ACM Trans. Math. Software (TOMS), 9, 302–325.

Freund, R. W., 1992. Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices. SIAM J. Sci. Stat. Comput. 13(1), 425–448.

Golub, G. H., Van Loan, C. F., 1996. Matrix Computations. Johns Hopkins University Press, 694 pp.

Gould, N. I. M., Scott, J. A., Hu, Y., 2007. A numerical evaluation of sparse direct solvers for the solution of large sparse symmetric linear systems of equations. ACM Trans. Math. Software 33(2), 10.

Gupta, A., 2002. Improved symbolic and numerical factorization algorithms for unsymmetric sparse matrices. SIAM J. Matrix Analysis Appl., 24, 529–552.

Gupta, A., 2007. A shared-and distributed-memory parallel general sparse direct solver. Appl. Algebra Eng. Commun. Comput., 18, 263–277.

Gupta, A., Joshi, M., Kumar, V., 1998. WSMP: A high-performance shared- and distributed-memory parallel symmetric sparse linear solver. In PARA'98 Workshop on Applied Parallel Computing in Large Scale Scientific and Industrial Problems, B. Kagstrom, J. J. Dongarra, E. Elmroth, and J. Wasniewski, Eds., Springer Verlag. Also available as RC 21886, IBM T. J. Watson Research Center, Yorktown Heights, NY.

Gupta, A., Karypis, G., Kumar, V., 1997. Highly scalable parallel algorithms for sparse matrix factorization. IEEE Trans. Parallel Distrib. Syst. 8(5), 502–520.

Haber, E., Ascher, U. M., Aruliah, D. A., Oldenburg, D. W., 2000. Fast simulation of 3D electromagnetic problems using potentials. J. Comput. Phys. 163(1), 150–171.

Hiptmair, R., 1998. Multigrid method for Maxwell's equations. SIAM J. Numer. Anal. 36, 204–225.

Hu, W., Abubakar, A., Habashy, T. M., 2009. Joint electromagnetic and seismic inversion using structural constraints. Geophysics 74(6), R99–R109.

Jackson, J., 1999. Classical electrodynamics, 3rd ed. John Wiley & Sons.

Jahandari, H., Farquharson, C. G., 2015. Finite-volume modelling of geophysical electromagnetic data on unstructured grids using potentials. Geophys. J. Int. 202(3), 1859–1876.

Karypis, G., Kumar, V., 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. 20(1), 359–392.

Kelbert, A., Meqbel, N., Egbert, G. D., Tandon, K., 2014. ModEM: a modular system for inversion of electromagnetic geophysical data. Comput. Geosci. 66, 40–53.

Key, K., 2012. Marine electromagnetic studies of seafloor resources and tectonics. Surv. Geophys. 33, 135–167.

Kordy, M., Wannamaker, P., Maris, V., Cherkaev, E., Hill, G., 2016. 3-D magnetotelluric inversion including topography using deformed hexahedral edge finite elements and direct solvers parallelized on SMP computers – Part I: forward problem and parameter Jacobians. Geophys. J. Int. 204(1), 74–93.

Koric, S., Lu, Q., Guleryuz, E., 2014. Evaluation of massively parallel linear sparse solvers on unstructured finite element meshes. Comput. Struct. 141, 19–25.

Kuzmin, A., Luisier, M., Schenk, O., 2013. Fast methods for computing selected elements of the Greens function in massively parallel nanoelectronic device simulations. In F. Wolf, B. Mohr, and D. Mey, editors, Euro-Par 2013 Parallel Processing, Vol. 8097, Lecture Notes in Computer Science, pages 533–544, Springer Berlin Heidelberg.

Li, X. S., Demmel, J. W., 2003. SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems. ACM Transactions on Mathematical Software (TOMS) 29(2), 110–140.

Liu, J. W. H., 1992. The multifrontal method for sparse matrix solution: Theory and practice. SIAM Review 34(1), 82–109.

Newman, G. A., 2014. A review of high-performance computational strategies for modeling and imaging of electromagnetic induction data. Surv. Geophys., 35(1), 85–100.

Newman, G. A., Hoversten, G. M., 2000. Solution strategies for two-and three-dimensional electromagnetic inverse problems. Inverse Problems, 16(5), 1357–1375.

Operto, S., Virieux, J., Amestoy, P., L'Excellent, J.-Y., Giraud, L., Ali, H. B. H., 2007. 3D finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: A feasibility study. Geophysics 72(5), SM195–SM211.

Pardo, D., Paszynski, M., Collier, N., Alvarez, J., Dalcin, L., Calo, V. M., 2012. A survey on direct solvers for Galerkin methods. SeMA Journal 57(1), 107–134.

Pellerin, L., 2002. Applications of electrical and electromagnetic methods for environmental and geotechnical investigations. Surv. Geophys. 23(2–3), 101–132.

Pratt, R. G., Shin, C., Hick, G. J., 1998. Gauss–Newton and full Newton methods in frequency–space seismic waveform inversion. Geophys. J. Int. 133, 341–362.

Puzyrev, V., Cela, J. M., 2015. A review of block Krylov subspace methods for multisource electromagnetic modelling. Geophys. J. Int. 202, 1241–1252.

Puzyrev, V., Gutierrez, N., Rodriguez, J. E., Hanzich, M., de la Puente, J., 2015. Electromagnetic modeling using a massively parallel software framework. In 77th EAGE Conference & Exhibition.

Puzyrev, V., Koldan, J., de la Puente, J., Houzeaux, G., Vazquez, M., Cela, J. M., 2013. A parallel finite-element method for three-dimensional controlled-source electromagnetic forward modeling. Geophys. J. Int. 193, 678–693.

Rodi, W., Mackie, R. L., 2001. Nonlinear conjugate gradients algorithm for 2-D magnetotelluric inversion. Geophysics 66(1), 174–187.

Saad, Y., 2003. Iterative methods for sparse linear systems. SIAM Philadelphia.

Schenk, O., Gärtner, K., 2004. Solving unsymmetric sparse systems of linear equations with PARDISO. Fut. Gen. Comput. Systems 20, 475–487.

Siemon, B., Christiansen, A. V., Auken, E., 2009. A review of helicopter-borne electromagnetic methods for groundwater exploration. Near Surf. Geophys. 7(5–6), 629–646.

Siripunvaraporn, W., Egbert, G., Lenbury, Y., Uyeshima, M., 2005. Three-dimensional magnetotelluric inversion: data-space method. Phys. Earth Planet. Inter. 150(1), 3–14.

Sourbier, F., Haidar, A., Giraud, L., Ben-Hadj-Ali, H., Operto, S., Virieux, J., 2011. Three-dimensional parallel frequency-domain visco-acoustic wave modelling based on a hybrid direct/iterative solver. Geophys. Prospect., 59(5), 834–856.

Streich, R., 2009. 3D finite-difference frequency-domain modeling of controlled-source electromagnetic data: Direct solution and optimization for high accuracy. Geophysics 74, F95–F105.

Streich, R., 2015. Controlled-source electromagnetic approaches for hydrocarbon exploration and monitoring on land. Surv. Geophys., 1–34.

Tarantola, A., 2005. Inverse Problem Theory and Methods for Model Parameter Estimation. SIAM, 352 pp.

Zehner, B., Börner, J. H., Görz, I., Spitzer, K., 2015. Workflows for generating tetrahedral meshes for finite element simulations on complex geological structures. Comput. Geosci. 79, 105–117.

Zhdanov, M. S., 2009. New advances in regularized inversion of gravity and electromagnetic data. Geophys. Prospect., 57(4), 463–478.

Figure captions.

Figure 1. Illustration of an unstructured 3D mesh for EM modeling. Different types of EM signal, natural or artificial, interact with conductive or resistive targets and cause substantial responses that can be measured at the surface.

Figure 2. Real model (top) and the results of the unconstrained standalone EM inversion (middle) and constrained by the seismic results (bottom). Color represents electric conductivity at logarithmic scale.

Figure 3. PARDISO performance on a sequence of FD linear systems. Computational time (top), memory (middle) and performance (bottom).

Figure 4. PARDISO factorization performance on FD linear systems (blue line with markers) and matrix condition number (red line) for a range of decreasing frequencies.

Figure 5. Computational time of forward/backward solves for the M1 system.

Figure 6. Scalability of MUMPS and WSMP for the factorizations of the three largest systems.

Figure 7. Average memory per node.

Figure 8. Parallel speedup of both solvers.

Figure 9. WSMP factorization performance.

Table 1. Direct solvers included in this survey

(MF – multifrontal, SN - supernodal)

Name	Version	Method	Implementation	Academic license
MUMPS	5.0	MF	*Distributed* (MPI, limited OpenMP)	Free
PARDISO	5.0	SN	*Shared* (OpenMP, limited MPI)	Free, user-locked for 1 year
WSMP	15.01	MF	*Distributed* (MPI, OpenMP)	Free, limited for 90 days

Table 2. Test matrix characteristics.

Name	Method	Size	NNZ	Cond. number
M1	FE	1,012,432	31,756,326	4.9E+06
M2	FD	2,066,700	25,301,576	2.8E+08
M5	FE	5,016,912	168,239,000	2.0E+12
M8	FD	7,810,110	97,668,202	~1.0E+14

Table 3. Direct and iterative solution of the M1 system. RHS is the solution time for a single right-hand-side.

Solver	Wall clock time (sec)			Memory (GB)
	Preprocess	Factorization	RHS	
MUMPS	19	1170	2.5	33
PARDISO	25	1296	3	22
WSMP	22	810	2.5	41
Iterative	33		65	1.4

Table 4. Direct and iterative solution of the M2 system.

Solver	Wall clock time (sec)			Memory (GB)
	Preprocess	Factorization	RHS	
MUMPS	35	3725	4.5	51
PARDISO	52	4856	5.5	45
WSMP	21	1405	4	69
Iterative	36		120	1.5

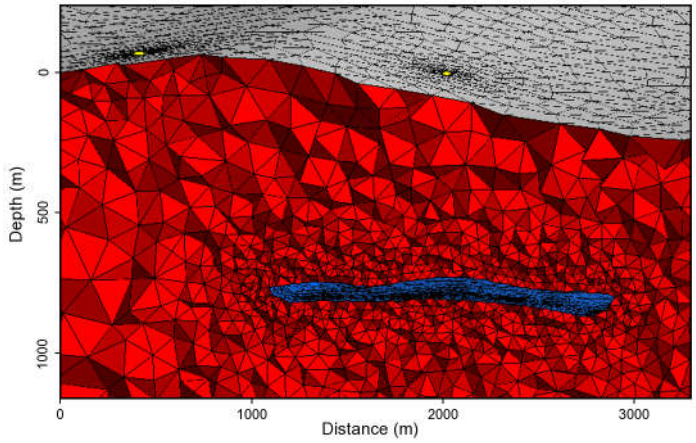

Figure 1.

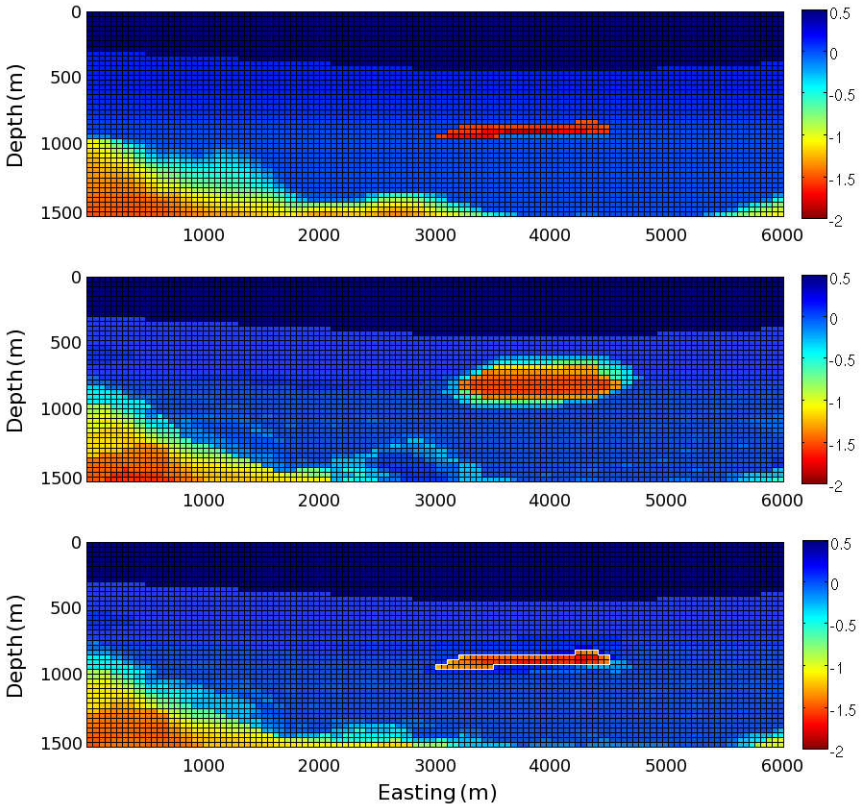

Figure 2.

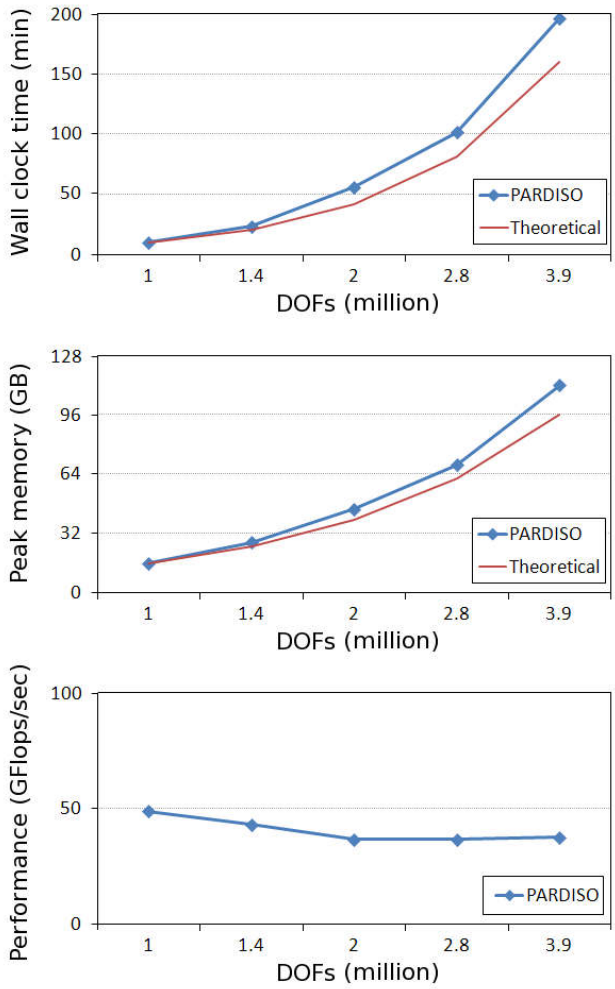

Figure 3.

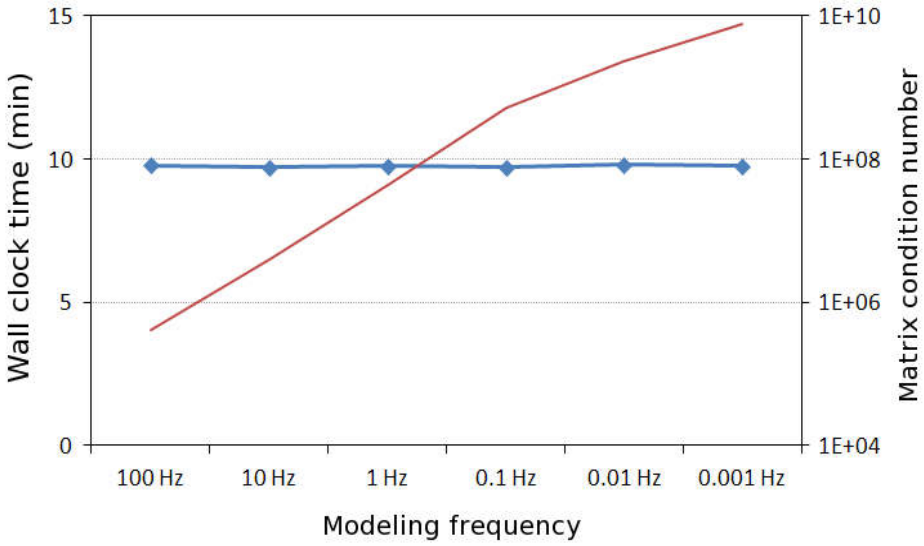

Figure 4.

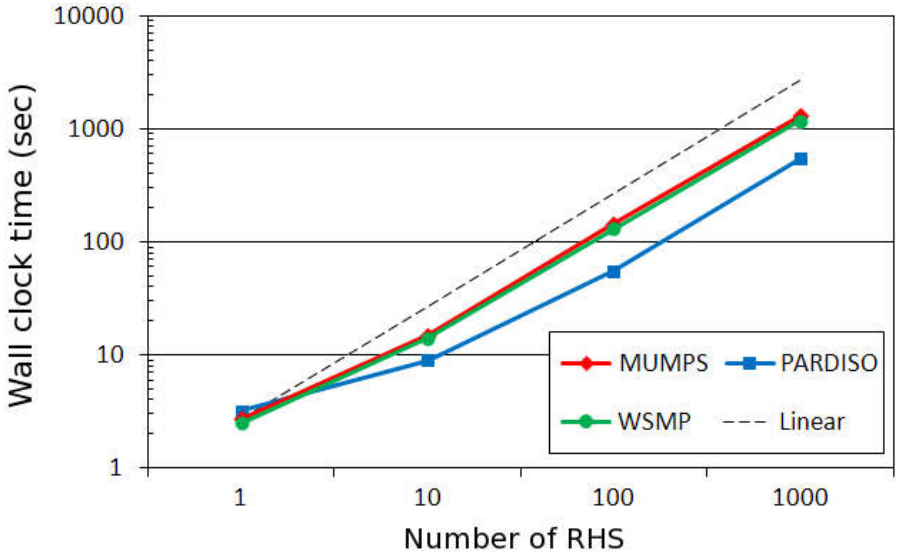

Figure 5.

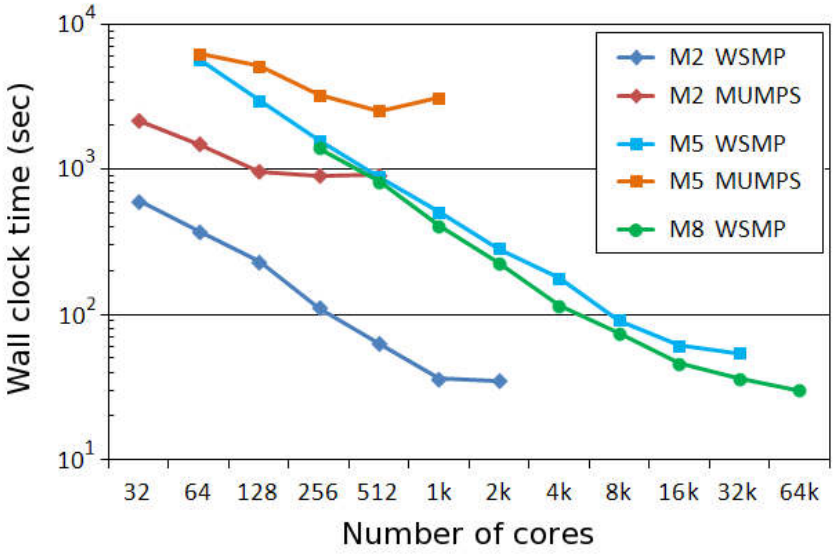

Figure 6.

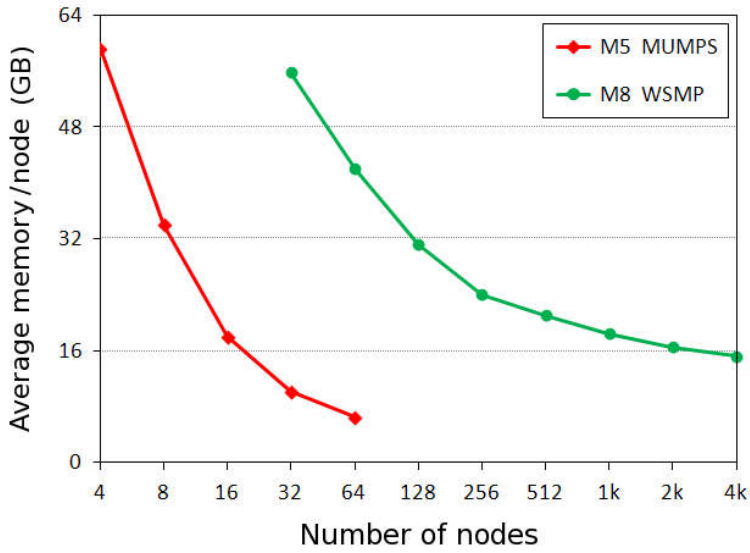

Figure 7.

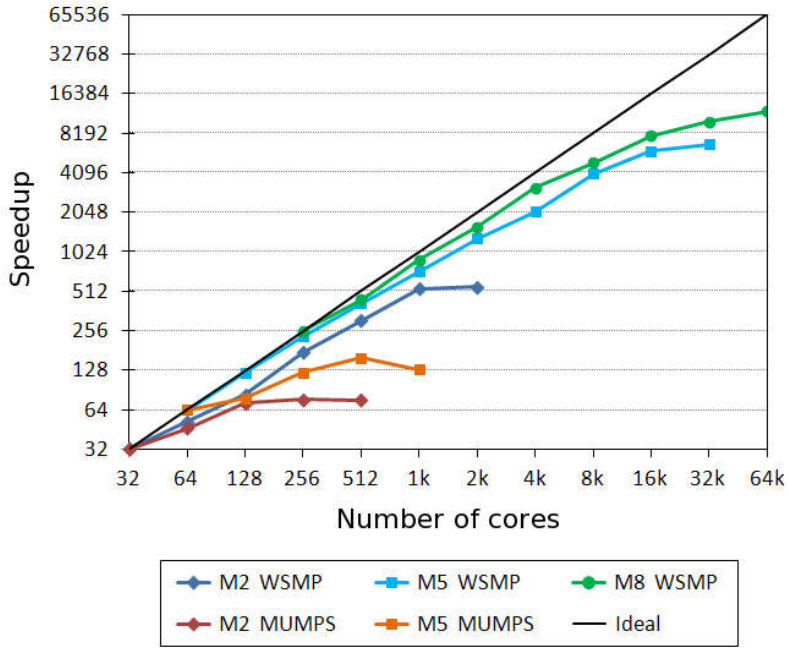

Figure 8.

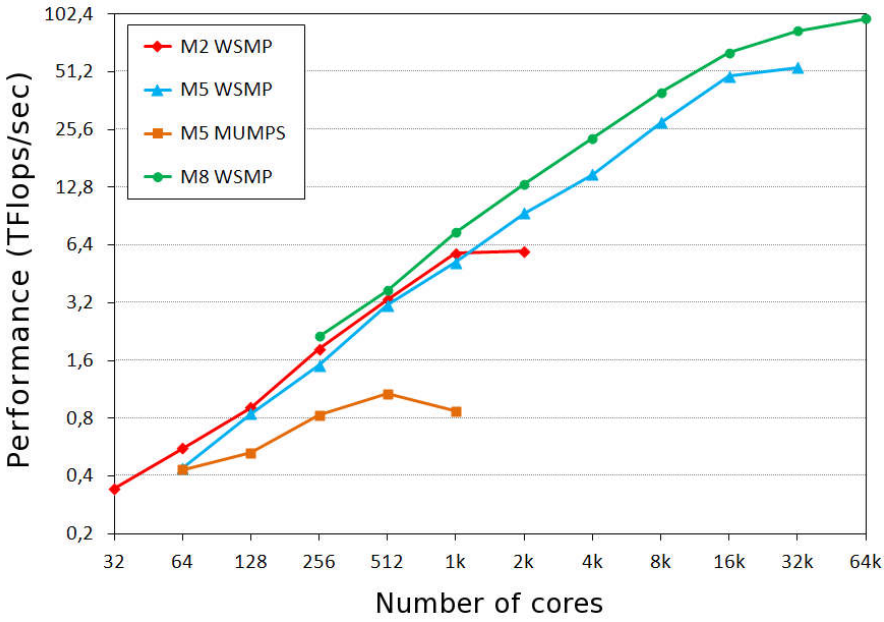

Figure 9.